

PCT/CN03/00230

# 证 明

REC'D 21 MAY 2003

WIPO PCT

本证明之附件是向本局提交的下列专利申请副本

申 请 日： 2002 03 29

申 请 号： 02 1 03824.4

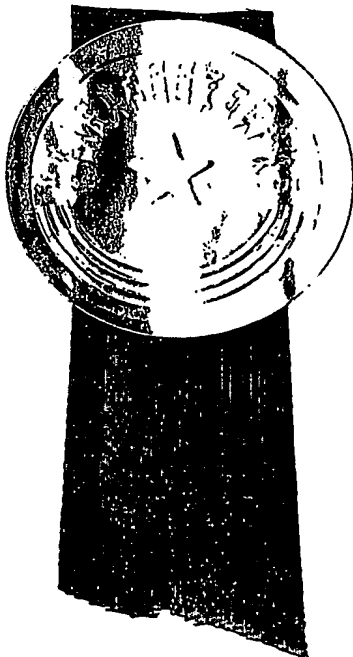
申 请 类 别： 发明

发明创造名称： 嵌入式应用以笔划中心线为基准描述的曲线字库

申 请 人： 上海汉峰信息科技有限公司

发明人或设计人： 曾冬初

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)



中华人民共和国  
国家知识产权局局长

王景川

2003 年 4 月 28 日

## 权 利 要 求 书

一种嵌入式应用以笔划中心线为基准描述的曲线字库，其特征在于：

1. 一种基于笔划中心线的笔划库，每一个笔划根据其形状分为若干笔划段，分别为笔划头段、正常段、拐段和为尾段，正常段和拐段可以有若干段，先用直线或贝塞尔曲线描述该笔划的中心线的各笔划段，又以该笔划中心线各段的关键点为各笔划段的控制点，各笔划段以各段控制点为基准的相对位置描述各笔划段的轮廓曲线；笔划数据为不等长，用尾段数据位标示该笔划结束；

笔划库的数据格式为：

字体所含笔划总数：占 2 个字节；

笔划头数据：笔划总数\*4 得到的字节，每个笔划占用 4 个字节，前 3 个字节为各笔划数据位置指针，后一个字节描述曲率变化的占 4 位的方式和占 4 位的段号；

笔划数据：含笔划各段中心线数据、笔划各段轮廓线数据。

每种不同的字体笔划数不同，一般在 1000 个左右，每个笔划数据的长度一般在 20~100 字节左右。

2. 一种由笔划直接构造的字库，其数据格式为：

相对于文件头的笔划库数据起始位置：4 字节；

笔划库所含汉字或字符，总数：4 字节；

汉字或字符索引表：4 乘所含汉字或字符总数得到的字节，每个字占 4 字节，前一个字节为该字所用笔划数，后 3 个字节为构成该汉字的各笔划的描述参数数据的位置指针；

参数数据：8\*形成该汉字或字符的笔划数得到的字节，每个笔划参数数据为 8 个字节，分别为：占 2 字节笔划号、占 1 字节厚度系数、占 2 字节笔划位置、占 12 位的笔划放大缩小系数、占 12 位的笔划曲率变化增量；

笔划库：如前 1 中所描述笔划库数据格式；

3. 一种由笔划构造的汉字部件库，其数据格式为：

部件库所含最多笔数构造的部件的笔划：4 字节，一般为 20~29，即一个部件最多由 20~29 个笔划构成，部件的排列是按 1 笔划部件、2 笔划部件、直到 20~29 笔划部件依次排列；

N 笔划部件索引表：设最多笔划数为 N，各笔划数所构造的部件数据的位置指针占 4\*N 字节，每种笔划数部件位置占 4 字节，分别为 1 笔划部件位置、2 笔划部件位置直到 N 笔划部件位置，每笔划数部件有若干个，从 1 号到到该笔划数部件总数为止排序，每个 2 笔划部件参数数据占 2\*8 个字节，每个 3 笔划部件参数数据占 3\*8 个字节，依此类推；

参数数据：8\*形成该部件的笔划数得到的字节，每个笔划参数数据为 8 个字节，分别为：占 2 字节的笔划号、占 2 字节的厚度系数、占 2 字节的笔划位置、占 12 字节的笔划放大缩小系数、占 12 字节的笔划曲率变化增量；

4. 一种由部件生成的汉字库，其数据格式为：

版本信息：4 字节，前 2 字节为公司标志简称，第三字节为字体标志，第四字节字符集标志；

字参数区起始位置：4 字节；

部件参数区起始位置：4 字节；

笔划数据区起始位置：4 字节；

字索引数据表：4\*总字数加 1 的和得到的字节，第一个 4 字节为总字数，以后每 4 字节表示每个字的索引，其中前 3 个字节表示字参数数据位置，后 1 字节为构成该字的部件数；

字参数数据：描述构成字的每一个部件的参数占 6 个字节，分别为占 11 位的部件号、该部件所含占 5 位的笔划数、占 2 字节的部件位置、占 12 位的部件缩小放大系数、占 6 位的部件的厚度系数；

部件参数数据：参见 3；笔划库：参见 1；

5. 多字体紧凑共享格式字库，各种字体构成每一个字的部件号是相同的，各种字体构成部件的笔划号是相同的，这一部分数据只要在一种基本字体中描述，而其他字体共享这部分数据，其数据格式描述如下：

基本字体的格式已在一、4 中描述；

其他字体的数据格式描述如下：

字参数区起始位置：4 字节；

部件参数区起始位置：4 字节；

笔划数据区起始位置：4 字节；

字索引数据表：4\*（总字数加 1 的和得到的字节，第一个 4 字节为总字数，以后每 4 字节表示每个字的索引，其中前 3 个字节表示字参数数据位置，后 1 字节为构成该字的部件数；

字参数数据：描述构成字的每一个部件的参数占 4 个字节，分别为、占 2 字节的部件位置、占 12 位的部件缩小放大系数、占 6 位的部件的厚度系数；

部件参数数据：参见 3，但去掉笔划号；笔划库：参见 1；

6. 以笔划中心线为基准描述的曲线字库之笔划还原解释器；
7. 以笔划中心线为基准描述的曲线字库之笔划组字还原解释器；
8. 以笔划中心线为基准描述的曲线字库之笔划组部件还原解释器；
9. 以笔划中心线为基准描述的曲线字库之部件组字还原解释器；
10. 以笔划中心线为基准描述的曲线字库之多字体部分共享数据的还原解释器。

# 说明书

## 嵌入式应用以笔划中心线为基准描述的曲线字库

### 技术领域:

本发明涉及一种由数字数据处理,尤其涉及一种用于将所要处理的数据转变成计算机能够处理的数据形式的字库,以用于计算机汉字信息处理。

### 背景技术:

字库是计算机操作系统的重要组成部分之一,因为汉字数量大,中文字符集大,字库则是中文信息处理的基础,在计算机上输入输出汉字、字符依赖字库。

传统的字库技术有点阵字库、矢量字库和轮廓曲线字库。

点阵字库有制作简单、还原解释程序简单等优点,但其缺点是不能放大、缩小(放大、缩小效果奇差),而且,大点阵字库的容量成几何倍数增加,一个应用需要多种不同尺寸规格的字库,所占的存储容量非常大,因而,点阵字库只是在早期的计算机系统和一些用字少、输出字形质量要求不高的简单的嵌入式专用设备上使用。

矢量字库和轮廓曲线字库是目前计算机使用的字库的主流,主要有 TTF 格式字库和 POSTSCRIPT 格式字库,目前在 PC 机市场上大部分使用这两种字库,其优点是技术成熟、字形优美、可放大缩小、可满足在 PC 以上的计算机上使用;其缺点是:数据格式复杂、字形还原解释器复杂、还原速度较慢、字体所占的存储容量大,不能满足其在嵌入式系统的上使用,且这两种格式的字库都是美国公司发明,其解释器不仅攫取我国大量金钱,而且妨碍我国对中文信息处理技术发展,我国在汉字字库技术领域受制于外国是不正常的。

### 发明内容:

首先,现在市场上还没有一种能够满足嵌入式系统使用的轮廓曲线字库,这是因为嵌入式系统一般存储容量小,增加存储器势必增加成本,一种 GB18030 字体的 TTF 格式字库需要 20M 左右的存储空间,用 4 种要 80M,而现在高档的嵌入式系统一般只有 16M 的存储空间,操作系统要占用一部分,给应用要留一部分,就根本不能放汉字字库,目前许多较高档的西文 PDA 或手机都因没有中文化不能投入到中国使用,因而,目前较高档的中文嵌入式系统就找不到合适的技术解决方案,因而,研制一个字形优美、可放大缩小、数据格式清晰、还原解释器清晰、字形还原速度快,特别是在上述前提下所占的存储容量小特别适于中文嵌入式系统的新型字库就非常必要了。

另外,我们必须研制一种中国自定义格式的字库,自行研制其中国自主知识产权的字形解释器,并将使其成为国家标准,形成外国公司的嵌入式产品进入中国的技术屏障,保障我国在嵌入式系统的中文信息处理应用技术领域的领先地位。

## 本发明的实现过程:

本发明首先使用标准 C 编程在 PC 机的 WINDOWS 操作系统上实现, 然后又已在嵌入式操作系统 NUCLEUS 上实现并固化在存储器上(试验产品), 并在不久的将来将其做成字库芯片和字库解释器芯片并固化在嵌入式 CPU 中。

本字库全名为“以笔划中心线为基准描述字的数据格式的系列字库产品”, 简称“嵌入式应用 ZGX 曲线字库”。为简便起见即用“ZGX”代替“以笔划中心线为基准描述的”。

1. 一种基于笔划中心线的笔划库, 每一个笔划根据其形状分为若干笔划段, 分别为笔划头段、正常段、拐段和为尾段, 正常段和拐段可以有若干段, 先用直线或贝塞尔曲线描述该笔划的中心线的各笔划段, 又以该笔划中心线各段的关键点为各笔划段的控制点, 各笔划段以各段控制点为基准的相对位置描述各笔划段的轮廓曲线; 笔划数据为不等长, 用尾段数据位标示该笔划结束;

笔划库的数据格式为:

字体所含笔划总数: 占 2 个字节;

笔划头数据: 笔划总数\*4 得到的字节, 每个笔划占用 4 个字节, 前 3 个字节为各笔划数据位置指针, 后一个字节描述曲率变化的占 4 位的方式和占 4 位的段号;

笔划数据: 含笔划各段中心线数据、笔划各段轮廓线数据。

每种不同的字体笔划数不同, 一般在 1000 个左右, 每个笔划数据的长度一般在 20~100 字节左右。

2. 一种由笔划直接构造的字库, 其数据格式为:

相对于文件头的笔划库数据起始位置: 4 字节;

笔划库所含汉字或字符, 总数: 4 字节;

汉字或字符索引表: 4 乘所含汉字或字符总数得到的字节, 每个字占 4 字节, 前一个字节为该字所用笔划数, 后 3 个字节为构成该汉字的各笔划的描述参数数据的位置指针;

参数数据: 8\*形成该汉字或字符的笔划数得到的字节, 每个笔划参数数据为 8 个字节, 分别为: 占 2 字节笔划号、占 1 字节厚度系数、占 2 字节笔划位置、占 12 位的笔划放大缩小系数、占 12 位的笔划曲率变化增量;

笔划库: 如前 1 中所描述笔划库数据格式;

3. 一种由笔划构造的汉字部件库, 其数据格式为:

部件库所含最多笔数构造的部件的笔划: 4 字节, 一般为 20~29, 即一个部件最多由 20~29 个笔划构成, 部件的排列是按 1 笔划部件、2 笔划部件、直到 20~29 笔划部件依次排列;

N 笔划部件索引表: 设最多笔划数为 N, 各笔划数所构造的部件数据的位置指针占 4\*N 字节, 每种笔划数部件位置占 4 字节, 分别为 1 笔划部件位置、2 笔划部件位置直到 N 笔划部件位置, 每笔划数部件有若干个, 从 1 号到到该笔划数部件总数为止排序, 每个 2 笔划部件参数数据占 2\*8 个字节, 每个 3 笔划部件参数数据占 3\*8 个字节, 依此类推;

参数数据: 8\*形成该部件的笔划数得到的字节, 每个笔划参数数据为 8 个字节, 分别为: 占 2 字节的笔划号、占 2 字节的厚度系数、占 2 字节的笔划位置、占 12 字节的笔划放大缩小系数、占 12 字节的笔划曲率变化增量;

4. 一种由部件生成的汉字库，其数据格式为：

版本信息：4 字节，前 2 字节为公司标志简称，第三字节为字体标志，第四字节字符集标志；

字参数区起始位置：4 字节；

部件参数区起始位置：4 字节；

笔划数据区起始位置：4 字节；

字索引数据表：4\*总字数加 1 的和得到的字节，第一个 4 字节为总字数，以后每 4 字节表示每个字的索引，其中前 3 个字节表示字参数数据位置，后 1 字节为构成该字的部件数；

字参数数据：描述构成字的每一个部件的参数占 6 个字节，分别为占 11 位的部件号、该部件所含占 5 位的笔划数、占 2 字节的部件位置、占 12 位的部件缩小放大系数、占 6 位的部件的厚度系数；

部件参数数据：参见 3；笔划库：参见 1；

5. 多字体紧凑共享格式字库，各种字体构成每一个字的部件号是相同的，各种字体构成部件的笔划号是相同的，这一部分数据只要在一基本字体中描述，而其他字体共享这部分数据，其数据格式描述如下：

基本字体的格式已在一、4 中描述；

其他字体的数据格式描述如下：

字参数区起始位置：4 字节；

部件参数区起始位置：4 字节；

笔划数据区起始位置：4 字节；

字索引数据表：4\*（总字数加 1 的和得到的字节，第一个 4 字节为总字数，以后每 4 字节表示每个字的索引，其中前 3 个字节表示字参数数据位置，后 1 字节为构成该字的部件数；

字参数数据：描述构成字的每一个部件的参数占 4 个字节，分别为、占 2 字节的部件位置、占 12 位的部件缩小放大系数、占 6 位的部件的厚度系数；

部件参数数据：参见 3，但去掉笔划号；笔划库：参见 1；

6. ZGX 字库之笔划还原解释器（程序模块）；

7. ZGX 字库之笔划组字还原解释器（接口应用程序模块）；

8. ZGX 字库之笔划组部件还原解释器（程序模块）；

9. ZGX 字库之部件组字还原解释器（接口应用程序模块）；

10. ZGX 字库之多字体部分共享数据的还原解释器（接口应用程序模块）。

本发明与现有技术相比的优点：

本发明与现有技术相比的优点在于：在保证字形的精美的前提下做到最省存储容量，一种字体的 GB18030 字库（含 27590 个汉字和 1136 个半角全角字符，共 28726 个编码）的标准格式字库文件为 1.1M 字节（而目前市场上使用的 TTF 格式字库 20M 字节左右）；本字库一种字体的紧凑格式占 800K 字节；四种字体共享紧凑格式字库：宋体为 800K，仿宋、黑体、楷体各为 600K，四种字体共 2600K 字节（2.6M）；七种字体为 4.5M 字节（而使用 TTF 格式需要超过 100M）；本发明的字形解释器清晰，特别优化，所占存储少，执行代码仅 40K 字节左右，还原速度快；本发明使用标准 C 编程，核心模块全部使用整数运算，易于与任何计算机操作系统接口；本发明字库除可任意缩放外，还增加了可无级改变字体笔

划粗细的功能，这是现有字库做不到的；特别在存储器资源有限的嵌入式应用中本发明技术可显示无可比拟的优越性，为较高档的中文嵌入式系统就找到了合适的不可替代的技术解决方案；因而本发明应用前景非常广阔。

附图说明：

图 1：主程序流程图，

图 2：将字库数据文件读入内存数组流程图，

图 3：从汉字字符的国际码或 Unicode 编码转换到 ZGX 字库的  
      字号（下列框图以国际码为例），

图 4：根据 ZGX 字库字号画笔划字流程图，

图 5：画笔划字流程图，

图 6：贝塞尔曲线递归调用程序，

图 7：用一条或多条贝塞尔曲线画一笔划数的子程序，

图 8：用一条或多条贝塞尔直线画一笔划数的子程序，

图 9：画直线转换子程序，

图 10：画直线子程序，

图 11：在内存缓冲区中画点位图中的点的子程序，

图 12：填充算法子程序，

图 13：建立边的分类表（E 表）子程序，

图 14：将该边的分类表（E 表）中的某一项的所有边插入 AE  
      活化边表中，并按 cxBotoom 升序排列。

## 具体实施方式:

按照图 1—14 所示来描述本发明的实施:

### 一、概述

本技术属计算机汉字信息处理范畴。本技术首先自定义一种独特的曲线轮廓字库数据格式,这种数据格式是依据本发明所定义的笔划和部件;然后根据这种数据格式使用解释程序还原所需要的精美的汉字字形;本发明的特点在于:在保证字形的精美的前提下做到最省存储容量,一种字体的 GB18030 字库(含 27590 个汉字和 1136 个半角全角字符,共 28726 个编码)的标准格式字库文件为 1.1M 字节(而目前市场上使用的 TTF 格式字库超过 15M);紧凑格式为 800K;四种字体共享紧凑格式字库:宋体为 800K,仿宋、黑体、楷体各为 600K,四种字体共为 2600K (2.6M);七种字体为 4.5M (而使用 TTF 格式需要超过 100M)。特别在存储器资源有限的嵌入式应用中本发明技术可显示无可比拟的优越性,因而应用前景非常广阔。

**ZGX** 字库使用的术语定义如下:(**ZGX** 代表以笔划中心线为基准描述的,以下同)

#### 1. **ZGX** 字库笔划

**ZGX** 字库笔划是构成字的最基本的元素。不同的笔划用笔划号来标示,汉字的笔划有横、竖、撇、捺、点等五大类,约 30 多种,但在 **ZGX** 字库上根据不同的需要,要制作 1000 多个不同的笔划。笔划由笔划中心线、笔划段构成,各笔划段用轮廓线描述,最后组成笔划的外轮廓闭合曲线。

笔划中心线是笔划的骨架线,它包括笔划各段的控制点,本身又由直线和贝塞尔曲线构成。

笔划段分为头段、正常段(主干段)、拐段、尾段,各段以笔划中心线定义的控制点为基准用相对于控制点的坐标来描述其各段的轮廓线,轮廓线由直线和贝塞尔曲线构成。

笔划段段号是指笔划正常段所属的段号,从 0 到最大的段号。

笔划段曲率变化方式是指头段、尾段、某正常段随方式的不同,其控制点坐标位置会发生变化。曲率变化方式有 15 种,从 1 到 15。

曲变段号和曲变方式将在笔划库的索引文件中给出。

#### 2. **ZGX** 字库部件

**ZGX** 字库部件是由若干笔划构成的,以所构成的笔划数来分类成:1 笔划部件、2 笔划部件、3 笔划部件...最多为 31 笔划部件;部件由元素(指几笔划部件)和在该元中的部件号来标示。在多字体字库产品中这些数据是可以共享的。

组成部件的各笔划的参数数据由笔划号、笔划厚度系数、笔划中心在部件坐标系下的坐标、笔划在 x 方向和 y 方向上的缩放系数、笔划在 x 方向和 y 方向上的曲率变化增量构成。

组成部件的笔划号在多字体字库产品中这些数据是可以共享的。

笔划中心在部件坐标系下的坐标限制在-128 到+127,用一个字节表示,这些数据是在定制字库产品时确定的。

笔划厚度系数作用于笔划的轮廓曲线,厚度系数的不同会使笔划有不同的粗细,



也会改变特殊段，从而改变字体的修饰。

笔划在 x 方向和 y 方向上的缩放系数只作用于笔划中心线的各控制点，而不影响笔划的粗细，从而保障了字体笔划粗细的一致性；笔划在 x 方向和 y 方向上的缩放系数也是在定制字库产品时确定的，限制在 0 到 255，各用一个字节表示，128 表示缩放系数为 1 倍；在紧凑格式中也可以用 6 位表示，限制在 32 到 95 之间，存放格式为上述数字减 32，故在 0 到 63 之间，64 表示缩放系数为 1 倍。

笔划在 x 方向和 y 方向上的曲率变化增量是指笔划头段、尾段和某正常段的曲率变化，曲率变化方式有 15 种，分别同时作用于头、尾和正常段中的一段或两段（最多两段），例如：头段的 x 坐标增量或尾段的 y 坐标增量是其中一种曲变方式，某正常段曲率 x 坐标增量和曲率 y 坐标增量也是其中一种曲率变化方式。前者解决复合笔划（例如：横折）的横段和折段的不同比例的问题，后者解决笔划在不同的部件中曲率不同的问题。引入曲率变化增量使 ZGX 字库的字形可以完全精确的逼近任意字体模板，从而保证了字库字形的精确和精美。曲率变化增量在 -128 到 127 之间，各用一个字节表示，在紧凑格式中曲率变化增量可以用 6 位格式表示，因为曲率变化增量一般较小。

### 3. ZGX 字库的字

ZGX 字库是用字号来标示，字号从 1 到 N(N 可以是一个长整数)，所以本字库对字的数目是没有限制，可以做成超大字库。ZGX 字库可以以 Unicode 编码排序，也可以以国标（GB2312、GBK、GB18030）编码来排序。

ZGX 字库的字是可以由若干部件构成，也可以直接由若干笔划构成，前者称为部件构字，后者称为笔划构字。

笔划构字是指字由若干个笔划构成，其参数数据由构成该字的每个笔划的笔划号、笔划厚度、笔划中心在字坐标系下的坐标、笔划在 x 方向和 y 方向的缩放系数、笔划在 x 方向和 y 方向上的曲率变化增量构成。笔划构字的各参数说明同一、2 中 ZGX 字库部件的描述。

部件构字是指字由若干个部件构成，其参数数据由组成该字的各部件所在的元数（几笔划部件）和部件号、该部件中心坐标在字坐标系下的坐标、该部件在 x 方向和 y 方向的缩放系数、该部件的厚度系数。而部件又由笔划构成，於一、2 中所述。

组成该字的各部件所在的元数（几笔划部件）和部件号在多字体字库产品中是可以共享的数据，只要在一种字体的字库文件里存在就可以了。

部件中心坐标在字坐标系下的坐标，该坐标是由该部件的结构位置号所确定的，结构位置号有 31 种，例如：左右结构、上下结构、包围结构等，这些是在定制字库产品时所确定的。

部件在 x 方向和 y 方向的缩放系数限定在 0 到 255 之间，各用一个字节表示，在紧凑格式中用 0 到 63 表示 32 到 95，用 6 位表示，缩放系数 64 表示为 1 倍。

### 4. 字生成外部调用参数

在设备坐标系下生成字，使用的参数如下：字的字体号、字的国标编码（或 Unicode 编码）、字的长度（以像素点为单位）、字的宽度（以像素点为单位）、字的厚度系数、字的前景色、字的显示方式（字的旋转、加粗、倾斜、填充方式等）；在缓冲区返回字的点位图数据。

字生成的外部调用参数中字的长度、宽度、厚度数据原则上没有限制，这就意味着一种字体可以以任意长度、宽度、厚度生成长体字、扁体字和任意粗细的字，一种字体可以作为数种字体使用。

## 5. ZGX 字库产品的技术指标

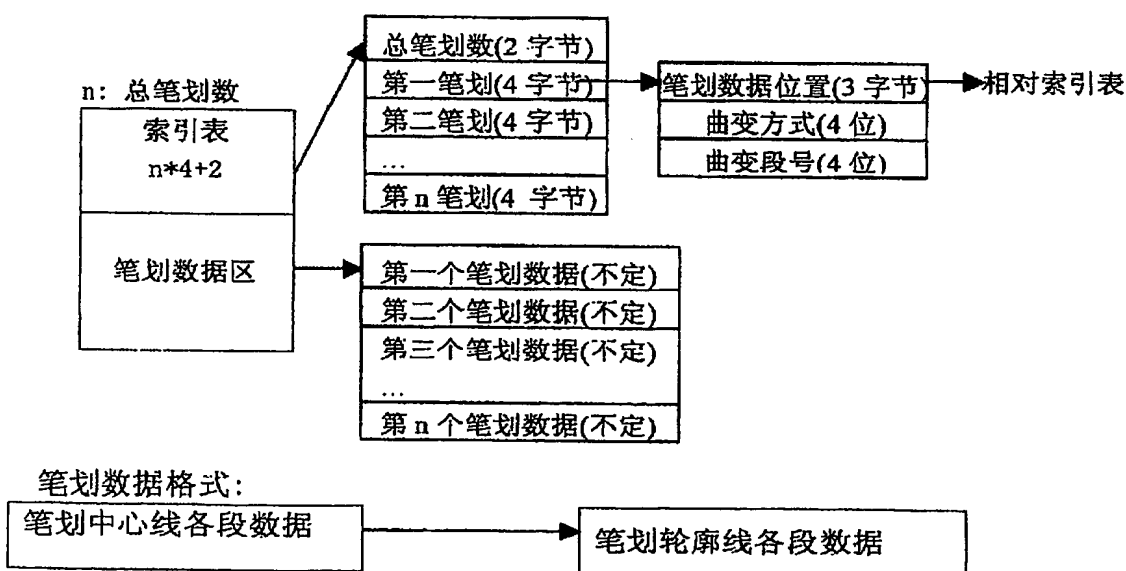
ZGX 字库在保证字形精确和精美的前提下，可以最大限度的节省存储容量和加快还原速度。

一种字体的 GB18030 字库（含 27590 个汉字和 1136 个半角全角字符，共 28726 个编码）的标准格式字库文件为 1.1M 字节（TTF 格式字库超过 15M）；紧凑格式为 800K；四种字体共享紧凑格式字库：宋体为 800K，仿宋、黑体、楷体各为 600K，四种字体共为 2600K（2.6M）；七种字体为 4.5M。

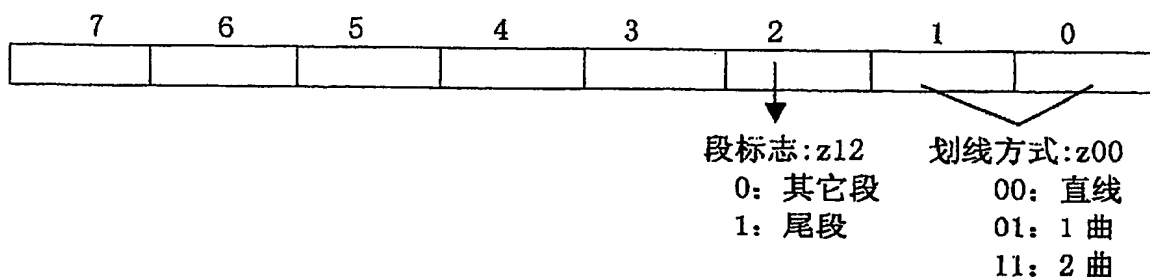
ZGX 字库的还原算法优于 TTF 字库，因为填充针对笔划进行，而不是整字，填充速度整体要快，加上 ZGX 字库索引简单直接，数学运算均采用整数运算并进行了全面优化。

## 二、 ZGX 字库数据格式

### 1. 笔划数据文件



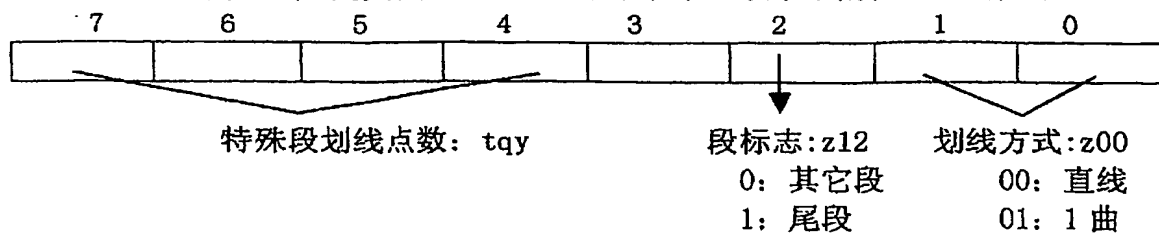
笔划中心线各段控制字数据格式，用一字节表示，该字节的位的约定如下：



笔划中心线各段数据：头段坐标（2 字节），段控制字（1 字节），正常段数据（ $2 \times z00$  字节）…，尾段控制字。

笔划轮廓线各段数据：

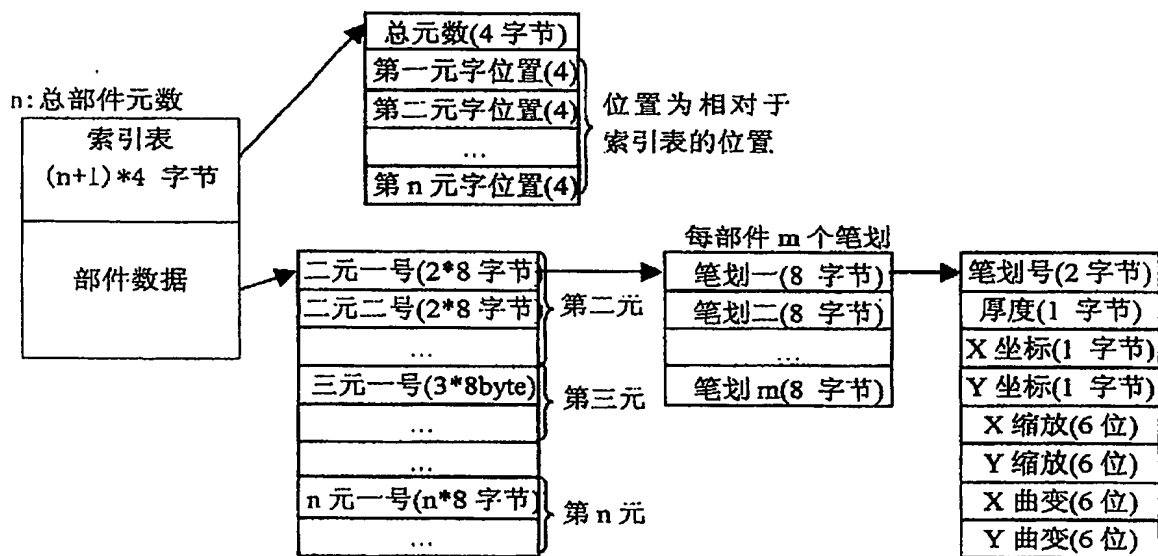
笔划轮廓线各段控制字数据格式，用一字节表示，该字节的位的约定如下：



笔划轮廓线各段数据：段控制字（1 字节），第一边轮廓线坐标增量（2 字节）

第二边轮廓线坐标增量（2 字节），头段其它点坐标（ $2 \times tqy$  字节），正常段数据（含第一、二边轮廓线各点坐标  $4 \times z00$  字节）；拐段控制字（1 字节），第一轮廓线坐标增量（2 字节），第二轮廓线坐标增量（2 字节），拐段其它点坐标（ $2 \times tqy$  字节），正常段数据（含第一、二轮廓线各点坐标  $4 \times z00$  字节）；…尾段控制字，尾段其它点坐标（ $2 \times tqy$  字节）。

## 2. 部件数据文件

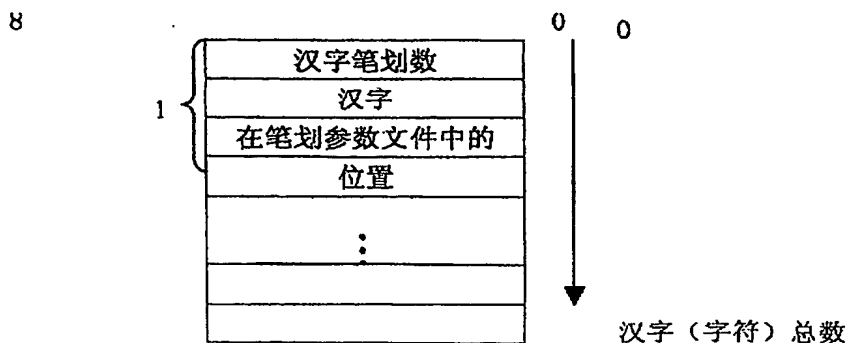


## 3. 笔划生成字的数据文件

前四字为笔划数据起始位置（相对于文件头）；

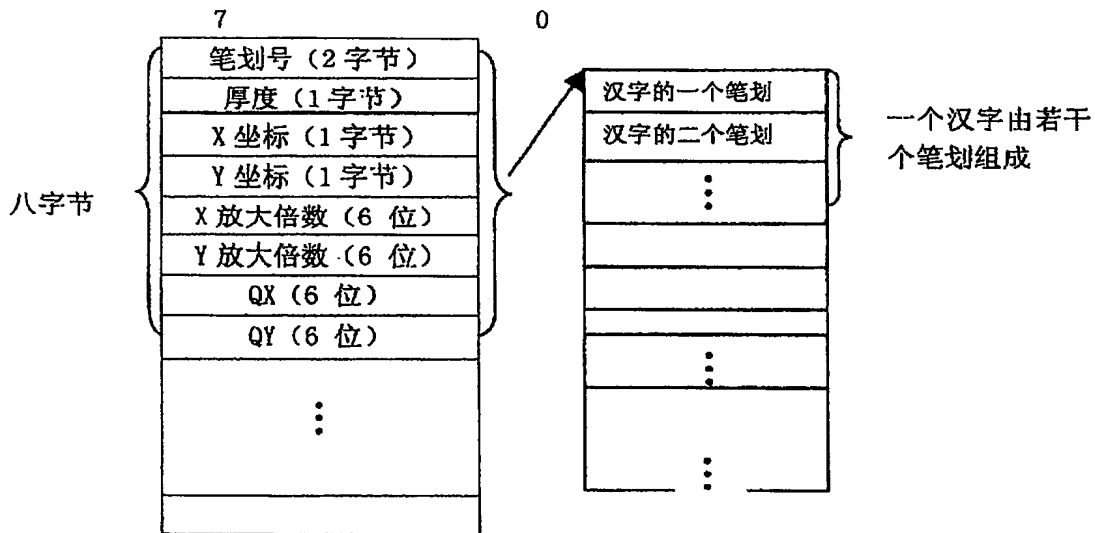
次四个字节为汉字（字符）总数；

接下来每个汉字参数数据索引占四个字节，按一定编码顺序排列，前一个字节为该汉字（字符）所含的笔划数，后三字为笔划数据在笔划参数文件中的位置，如下：



该数据块长度为  $4 \times \text{汉字(字符)总数}$ ;

接下来是笔划结构参数数据块: 每个笔划占八个字节, 第一、二字节为该笔划的笔划号, 第三字节为该笔划的厚度, 其后五字节分别为该笔划的 X、Y、SX、SY、QX、QY 如下所示:



该数据块存储容量为: 所有汉字(字符)所用笔划总数  $\times 8$  个字节;

接下来为笔划生成数据块, 组成如下:

总笔划数: 2 字节,

笔划数据索引:  $4 \times \text{总笔划数}$ , 笔划数据。(见二、1.(1) 笔划数据文件紧凑格式)

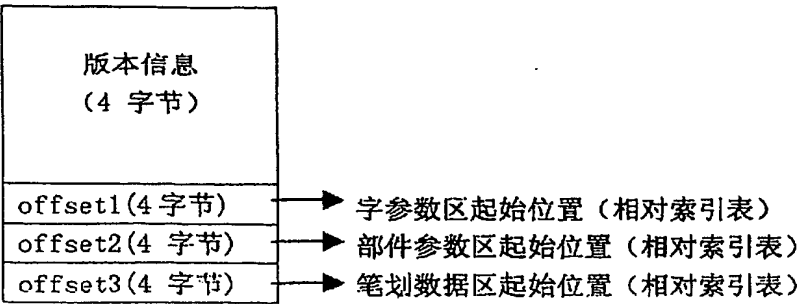
#### 4. 部件生成字数据格式

基本字体文件主要由以下部分组成:

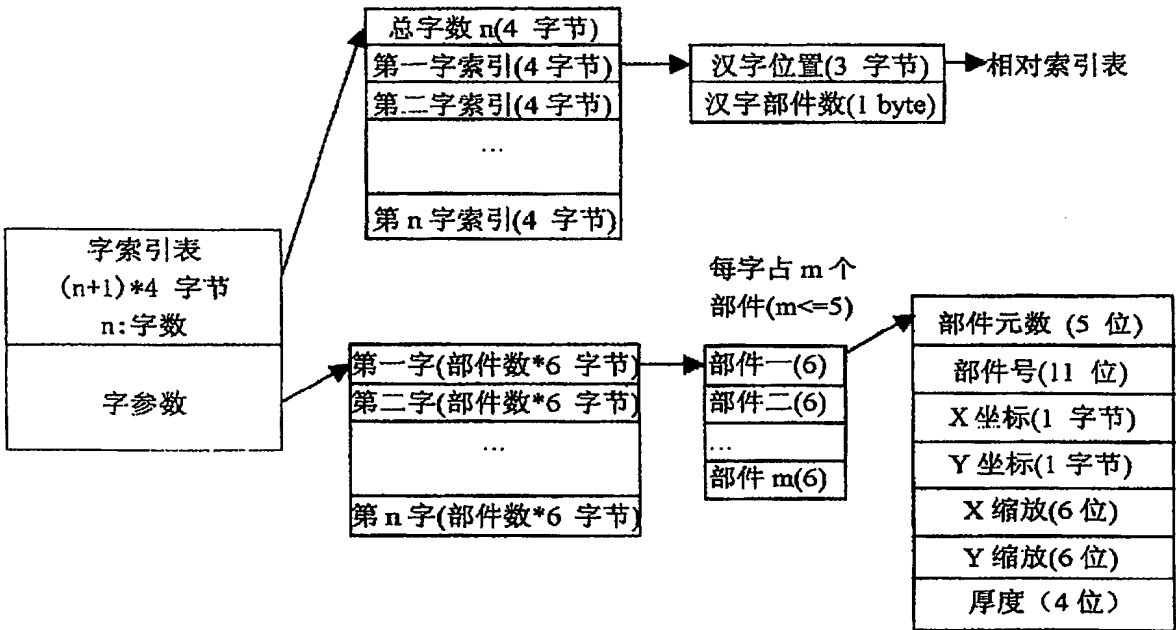
索引表
字参数
部件参数
笔划数据

各部分的详细数据格式描述如下：

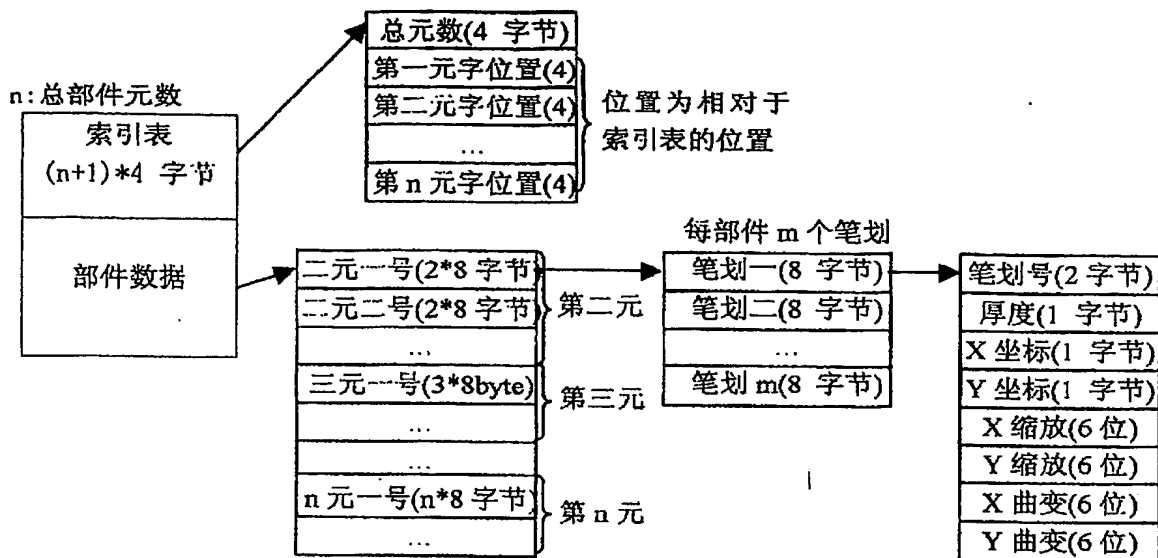
索引表：



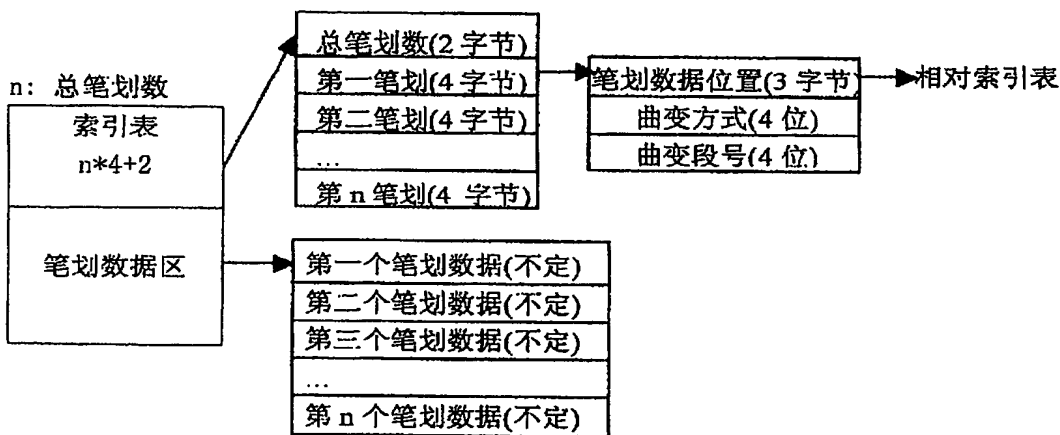
字参数数据



## 部件参数数据



## 笔划数据

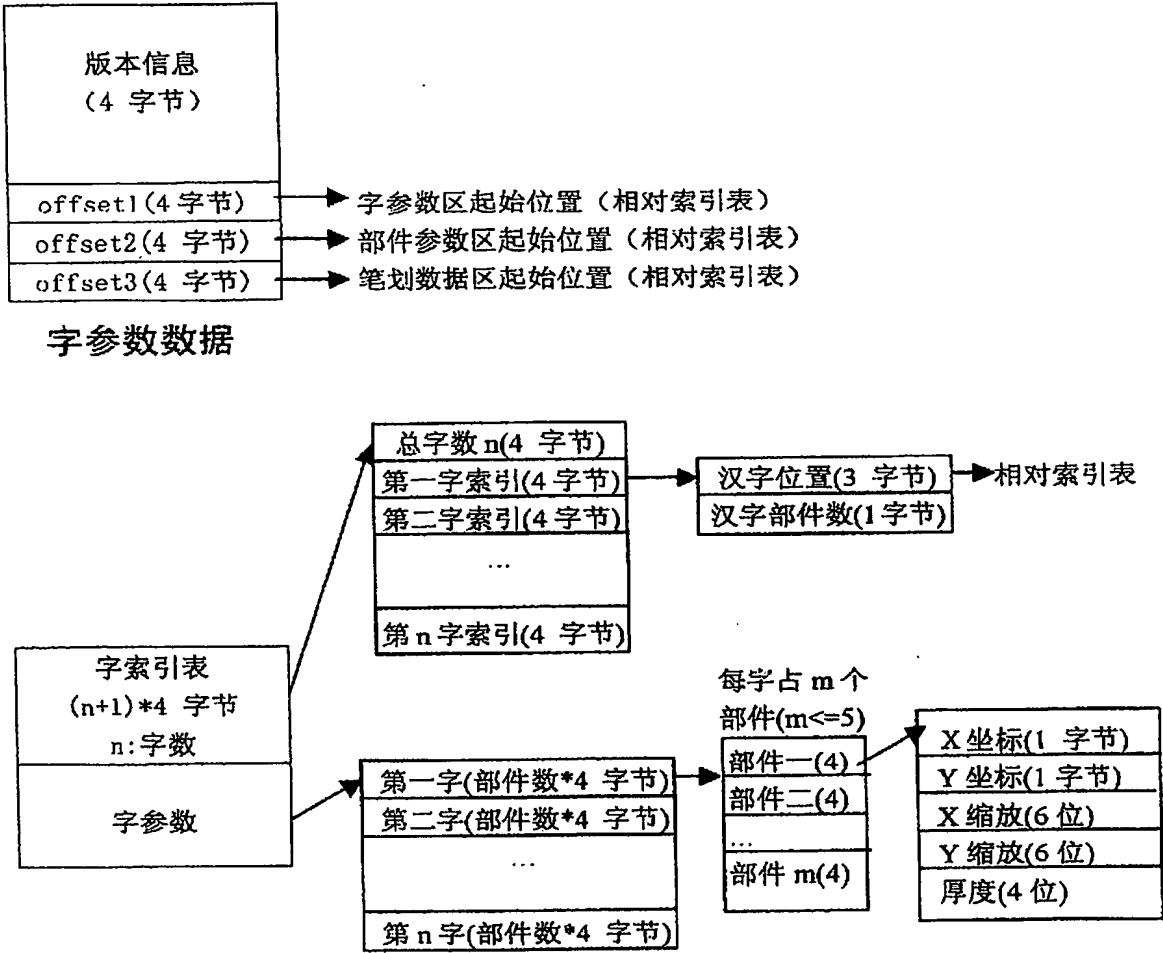


5. 多字体部分共享部件构字格式  
其他字体文件主要由以下部分组成:

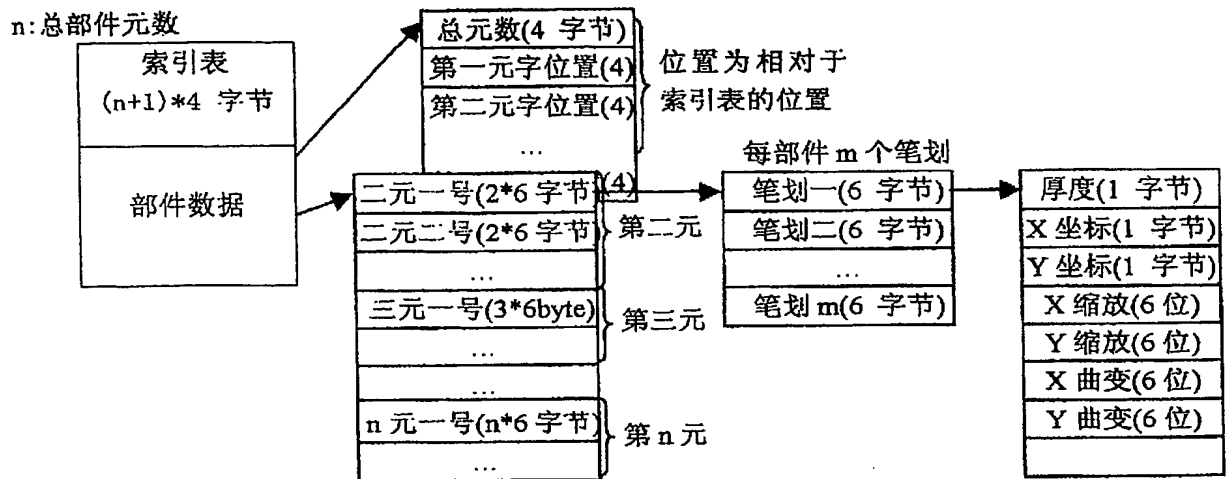
索引表
字参数
部件参数
笔划数据

各部分的详细数据格式描述如下：

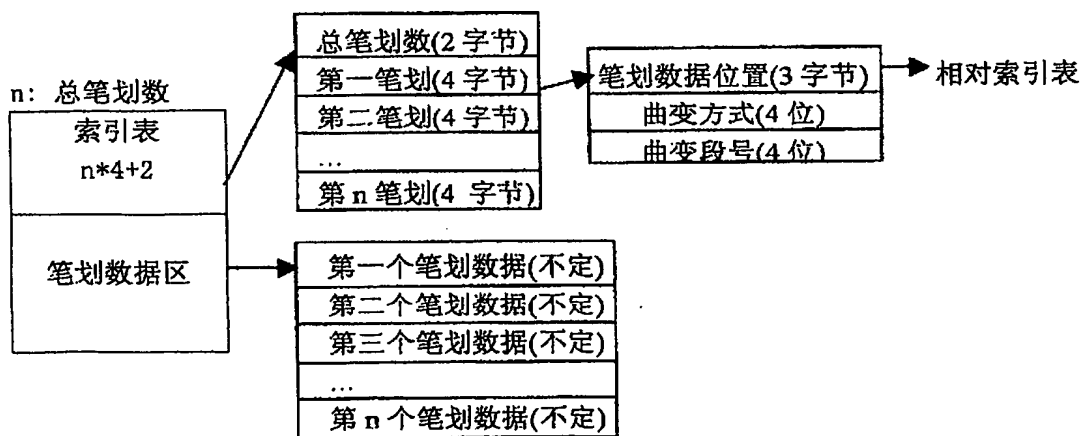
索引表：



## 部件参数数据



## 笔划数据





### 三、 数学模型和解释程序

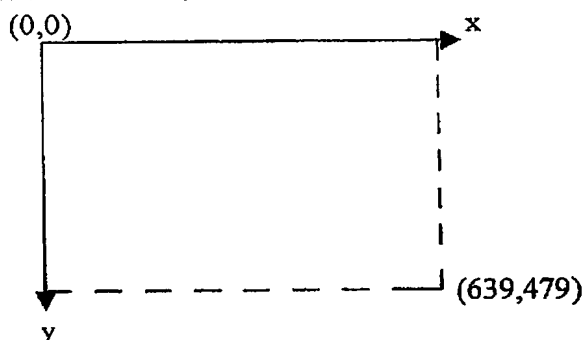
#### 1. 坐标系

ZGX 字库使用了以下各种坐标系：设备坐标系(对于显示打印或其它设备)、字坐标系、部件坐标系、笔划坐标系。

##### (1) 设备坐标系

以计算机显示器为例：屏幕坐标系定义如下图：

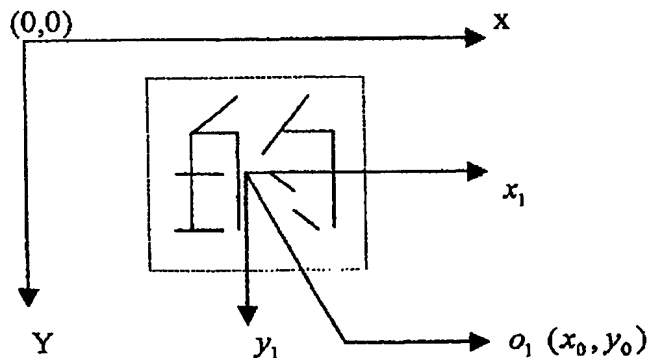
屏幕左上角坐标原点是(0,0)， 向下为 y 轴， 向右为 x 轴。VGA 显示方式分辨为  $640 \times 480$ ，因此屏幕右下角坐标为(639,479)。



##### (2) 字坐标系

该图  $x_1o_1y_1$  为坐标系，图中 xoy 坐标系是设备坐标系。在设备坐标系下为字的中心坐标，如图所示。

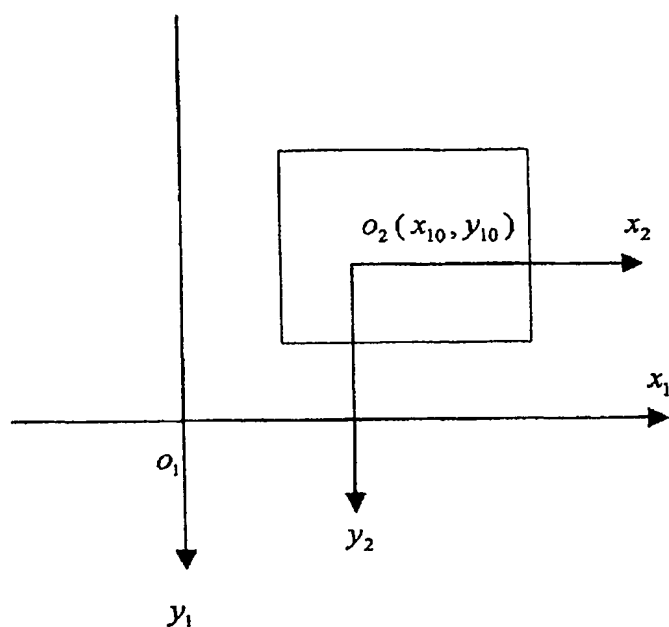
显示一个字，只要给出字号、x 方向和 y 方向的缩放系数(基准字为  $256 \times 256$  是分辨率)及字中心坐标  $o_1(x_0, y_0)$  (该点为字坐标系原点  $o_1$ )，就可以在屏幕的给定位置显示该字。



##### (3) 部件坐标系

如图  $x_2o_2y_2$  为部件坐标系；它是相对于字坐标下相对坐标系； $o_2$  为一字元的坐标中心；每一字元有一个相对与字坐标系下的部件坐标系；显示一个字是通过显

示若干部件实现的；显示该字只要从字结构文件中取出各部件号和各部件在  $x, y$  方向的缩放倍数及  $o_2$  点在坐标系的相对坐标  $(x_{10}, y_{10})$ 。

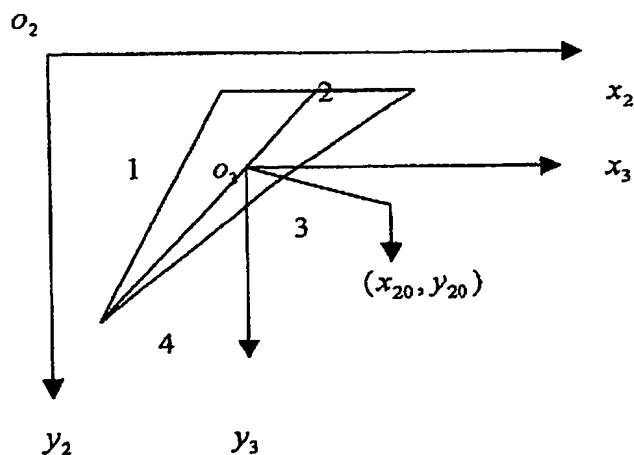


#### (4) 笔划坐标系

如图  $x_3oy_3$  为笔划坐标系；笔划坐标系是相对于部件坐标系  $x_2oy_2$  的相对坐标系。生成一个部件是通过显示若干笔划实现的；生成一个部件要从部件结构文件中取出：各笔划号及各笔划在  $x, y$  方向缩放倍数；

及  $o_3$  点在  $x_2oy_2$  坐标系下的相对坐标  $(x_{20}, y_{20})$  即可实现。同样，每一个笔划显示都有一个笔划坐标系。

如何画笔划呢，就是根据在笔划坐标下，该笔划的坐标点数及曲线拟和参数（根据相应笔划号从笔划文件中取出）采用填充算法获得。



#### (5) 贝塞尔二次曲线拟合

如图短撇分为 1、2、3 段，每段都用一个贝塞尔二次曲线拟合。

第一段使用①、②、③点坐标；其中第②段是第①、③段的切线的交点。

第二段使用③、④、⑤点坐标；其中第④是为第③、⑤ 的切线交点。

第三段使用⑤、⑥、①点坐标；其中第⑥是为第⑤、①的切线交点。

第①、③点连线为直线，第②点可在直线上任意点。

公式如下：

$$\begin{cases} x(t) = x_1(1-t^2) + 2t(1-t)x_2 + t^2x_3 & \text{-----(1)} \\ y(t) = y_1(1-t^2) + 2t(1-t)y_2 + t^2y_3 & \text{-----(2)} \end{cases}$$

其中  $t \in [0,1]$

式中  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$  分别为首端点、中间点及末端点的坐标。

这样贝塞尔曲线拟合也只要存储①到⑥点的相对坐标。与折线拟合的数据格式一致，所不同总是记录偶数个点的坐标值。且第偶数个点的值为前后两端点的切线交点。在填充计算时采用不同的方法。

(6) 在设备上生成一个字的坐标变换

每一个部件的每一个笔划的各坐标点由在字坐标系到设备坐标系的坐标变换公式如下：

$$X_{kij} = ((X_{2i} + S_{xBi} \times X_{3j} \div 128) \times B_{xk} \div 128 + X_{k1}) \times Z_{xs} \div 128 + X_0$$

$$Y_{kij} = ((Y_{2i} + S_{yBi} \times Y_{3j} \div 128) \times B_{yk} \div 128 + Y_{k1}) \times Z_{ys} \div 128 + Y_0$$

式中，

$(X_{kij}, Y_{kij})$  为 i 笔划各坐标点在字坐标系下的坐标，

$(Z_{xs}, Z_{ys})$  为整字的缩放系数，

$(X_0, Y_0)$  为字中心在屏幕坐标系下的坐标，

$(X_{3j}, Y_{3j})$  为 i 笔划各坐标点在笔划坐标系下的坐标，

$(X_{k1}, Y_{k1})$  为第 k 部件中心在字坐标下的坐标，

$(B_{xk}, B_{yk})$  为第 k 部件在 X 和 Y 方向上的缩放系数，

$(X_{2i}, Y_{2i})$  为第 k 部件的第 I 笔划中心在该字元坐标系下的坐标，

$(S_{xBi}, S_{yBi})$  为 k 部件的第 I 笔划在 X 和 Y 方向上缩放系数，

k=1-----该字元所含部件数;  
i=1-----第 k 部件所含的笔划数;  
j=1-----第 k 部件的第 i 笔划所含的点数;

## 2. ZGX 字库解释程序

程序流程图如下:

- (1) 主程序流程图, 见附图 1。
- (2) 将字库数据文件读入内存数组, 见附图 2。
- (3) 从汉字(字符)的国标码(或 Unicode 编码)转换到 ZGX 字库的字号(下列框图以国标码为例), 见附图 3。
- (4) 根据 ZGX 字库字号划笔划字, 见附图 4。
- (5) 根据 ZGX 字号划部件字, 见主程序流程图, 即附图 1。
- (6) 划笔划程序流程图, 见附图例 5。
- (7) 贝塞尔曲线递归调用程序, 见附图 6。
- (8) 用一条或多条贝塞尔曲线划一笔划段的子程序, 见附图 7。
- (9) 用一条或多条直线划一笔划段子程序, 见附图 8。
- (10) 划直线转换子程序, 见附图 9。
- (11) 划直线子程序, 见附图 10。
- (12) 在内存缓冲区中划点位图中的点的子程序, 见附图 11。
- (13) 填充算法子程序, 见附图 12。
- (14) 建立边的分类表(ET 表)的子程序, 见附图 13。
- (15) 将该边的分类表(ET)中某一项的所有边插入 AEL 活化边表中, 并按 `cxBottom` 升序排列, 见附图 14。

## 四、 ZGX 字库调用接口函数

### 1. 数据结构定义

#### (1) 字体初始化返回指针数组

```
typedef struct fhzz
{
    long * zys;
    long * zics;
    long * zrcs;
    long * bhcs;
} fhzz, far * lpfhzz;
zys: 字索引参数数据指针;
zics: 字体之字参数数据指针;
zrcs: 字体之部件参数数据指针;
zhcs: 字体之笔划参数数据指针。
```

#### (2) 汉字（字符）还原数组

```
typedef struct hzfycs
{
    unsigned char zhd;
    unsigned char zkd;
    unsigned char zcd;
    unsigned char zys;
    unsigned char zfs;
} hzfycs, far * lphzfycs;
zhd: 汉字的厚度参数, 1 字节, 标准为 32。
zkd: 汉字的宽度参数, 1 字节, 以象素点为单位。
zcd: 汉字的长度参数, 1 字节, 以象素点为单位。
zys: 汉字的颜色或灰度, 1 字节。
zfs: 汉字的显式方式参数, 1 字节;
```

参数值约定如下:

0 位和 1 位为旋转方式: 00 为常规 (不旋转), 01 为旋转 90 度,  
10 为旋转 180 度, 11 为旋转 270 度;

2 位为加粗与否: 0 为常规, 1 为加粗;

3 位为倾斜与否: 0 为常规, 1 为倾斜;

4 位为加下划线与否: 0 为常规, 1 为加下划线;

5 和 6 位为汉字填充方式: 00 为只划中心线即实现线体字, 01 为实现以或方式填充的汉字, 10 为实现以与方式填充的汉字; 11 为实现以异或方式填充的汉字;

7 位为字符显示方式, 0 为半角, 1 为全角, 汉字为 1。

## 2. 字体初始化

### (1) 接口函数

HZ\_S(zth, lpfhzz);

zth: 字体号, 字符变量, 此为入口参数;

参数值约定如下:

0: GB-2312 的 16\*16 点阵字库;

1: GB18030-2000 宋体(汉字)曲线字库;

2: GB18030 黑体(汉字)曲线字库;

3: GB18030 仿宋(汉字)曲线字库;

4: GB18030 楷体(汉字)曲线字库;

5: GB18030 字符曲线字库;

6: ...

Lpfhzz: 返回参数, 指针数组指针。

### (2) 实现方法:

根据字体号打开相应字体文件名, 例如 GB-2312 的 16\*16 点阵字库文件名为 DW16D, GB18030-2000 宋体曲线字库文件名为 DWSTQ, GB-2312 黑体曲线字库文件名为 DWHTQ, 字符库文件名为 DWQZFK;

定义 12 个内存数组分别为 GB-2312 的 16\*16 点阵字库数组, GB18030-2000 宋体曲线字库字索引参数数组、字体之字参数数组、部件参数数组和笔划参数数组, 黑体曲线字库字索引参数数组、字体之字参数数组、部件参数数组和笔划参数数组, 仿宋、楷体、字符库等的各参数数组;

将各字体文件数据读入相应的数组;

返回各数组指针。

## 3. 汉字(字符)还原

### (1) 接口函数

HZ\_X(hz\$, zth, hzfycs, fclx, zbuf);

Hzh: 字体号;

hz\$: 一个汉字(字符)的国标码内码, 通常为 2 字节。

hzfycs: 汉字的还原参数数组, 5 字节。

Fclx: 返回参数类型(字符型变量), 0 为点位图, 1 为轮廓矢量数组, 2 为 TTF 格式数据格式, 3 为 PS 格式数据格式。

Zbuf: 返回汉字(字符)点位图及其他类数据缓冲区指针; 此为返回参数;

汉字(字符)点位图数据存储方式为每象素点为 1 位, 每一个字所占字节数 =  $\lceil ((zkd * zcd) + 7) / 8 \rceil$ 。

### (2) 实现方法:

根据字体号找到相应数组指针;

根据汉字(字符)内码计算汉字字号, 计算公式如下: 即

设内码高字节为 C1, 低字节为 C2 (C1, C2 为无符号整数),

若  $C1=0$ ，则转半角字符处理；

当  $hz\$$  在字符编码的范围内，即但字国标编码在（ $0X1F-0X80$ 、 $A1A1-A1FE$  到  $A9A1-A9FE$ 、 $A840-A87E$  到  $A880-A8FE$  到  $A940-A97E$  到  $A980-A9FE$  共 1136 个编码）内时，调字符字库文件  $DWQZFK$ ，曲线字库为两级结构笔划构字，笔划还原子程序不用修改，只需改调用曲线字库文件，处理字符结构文件，根据字符编码换算出在字符库的顺序号  $ZH$ ：

```

if(hz$>=0x1f && hz$<=0x80) ZH=hz$-0x1f;
else if(c0>=0xa1 && c0<=0xa9 && c1>=0xa1 && c1<=0xfe)
ZH=99+(c0-0xa1)*94+c1-0xa1;
else if(c0>=0xa8 && c0<=0xa9 && c1>=0x40 && c1<=0xa0)
{
ZH=945+(c0-0xa8)*96+c1-0x40;
if(c1>0x7f) ZH--;
}

```

否则为汉字字库转相应字体文件先做如下处理：

若  $C1$  大于等于  $0XB0$  且小于等于  $0XF7$  且  $C2$  大于等于  $0XA1$  时（即 GB2312 汉字或双字节 2 区）：字号 =  $(C1-176) * 94 + C2 - 161$ ；

若  $C1$  大于等于  $0X81$  且  $C2$  小于  $0XA1$  时（即双字节 3 区）：当  $C2$  大于  $0X7F$  时  $C2--$ ，字号 =  $(C1-129) * 190 + C2 - 64 + 6768$ ；

若  $C1$  大于等于  $0XAA$  且  $C2$  小于  $0XA1$  时（即双字节 4 区）：当  $C2$  大于  $0X7F$  时  $C2--$ ，字号 =  $(C1-170) * 96 + C2 - 64 + 12848$ ；

四字节字字号为 21008-27589。

根据字号找到描述该字的数据，

若为  $16*16$  点阵字，该字的数据指针为：点阵字数据指针 + (字号 - 1) \* 32；

若为曲线字，则根据字索引文件和字号找到描述该字的字参数数据，然后根据字参数数据找到描述该字的各字元参数数据，然后根据字元参数数据找到描述各字元的各笔划数据。

计算各笔划数据的坐标。

调用划笔划子程序。

划笔划子程序调用划直线子程序和划贝塞尔曲线子程序，最后调用填充程序。

另外还有汉字（字符）的旋转、加粗、加下划线、倾斜子程序。

### (3) 设计

设计要求成一个接口程序和划笔划子程序、划直线子程序、划贝塞尔曲线子程序，填充子程序。

笔划画线和填充子程序要求在内存缓冲区内实现。

划字控制的计算都在  $256*256$  的矩形上，划中心线和轮廓线时再缩放（先均以字长和字宽为 256，在划笔划中心线和轮廓线时先将点的相对字中心坐标的  $X$  坐标 \* 字宽 / 256，相对字中心坐标的  $Y$  坐标 \* 字长 / 256）。小于等于  $20*20$  的字用线体字（指均小于，即只划中心线），等于  $16*16$  且为 GB2312 国标汉字用点阵字（其余字仍用线体字）。

返回为字的点位图、矢量轮廓、TTF 格式或 PS 格式。

00-03-29

容错，任何字在任何分辨率下显示不出错、不失真、不越界。

27

#### 4. 显示字符串

##### (1) 接口函数

HZ\_C(hzc\$, zth, zxx, zyy, hzfycs);

Hzh: 字体号;

hzc\$: 一个(或一组)汉字(字符)的内码字符串。

Zxx: 汉字(字符)串显示的左上角屏幕 X 坐标。

zyy: 汉字(字符)串显示的左上角屏幕 Y 坐标。

hzfycs: 汉字的还原参数数组, 5 字节。



说明书附图

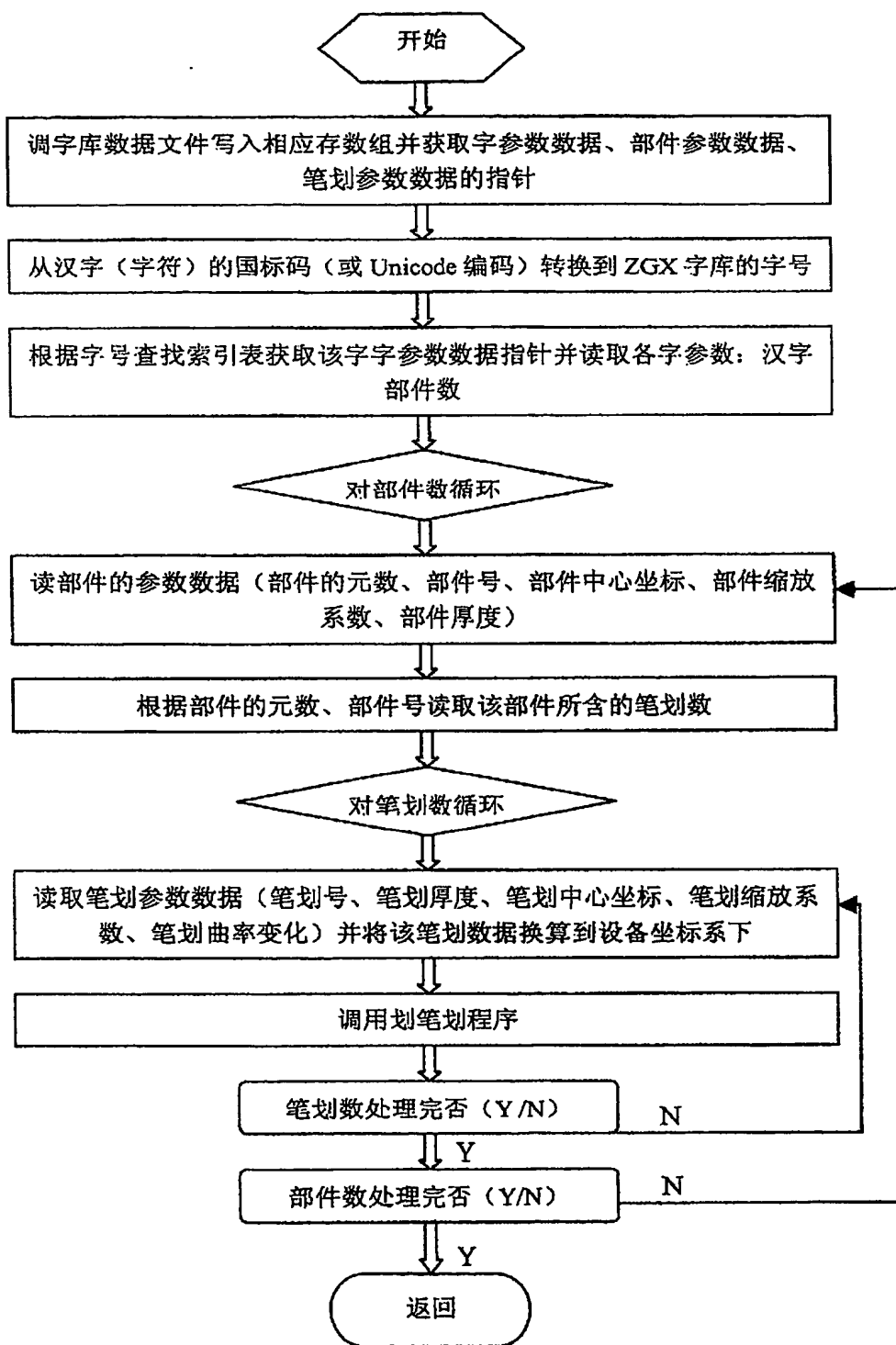


图 1

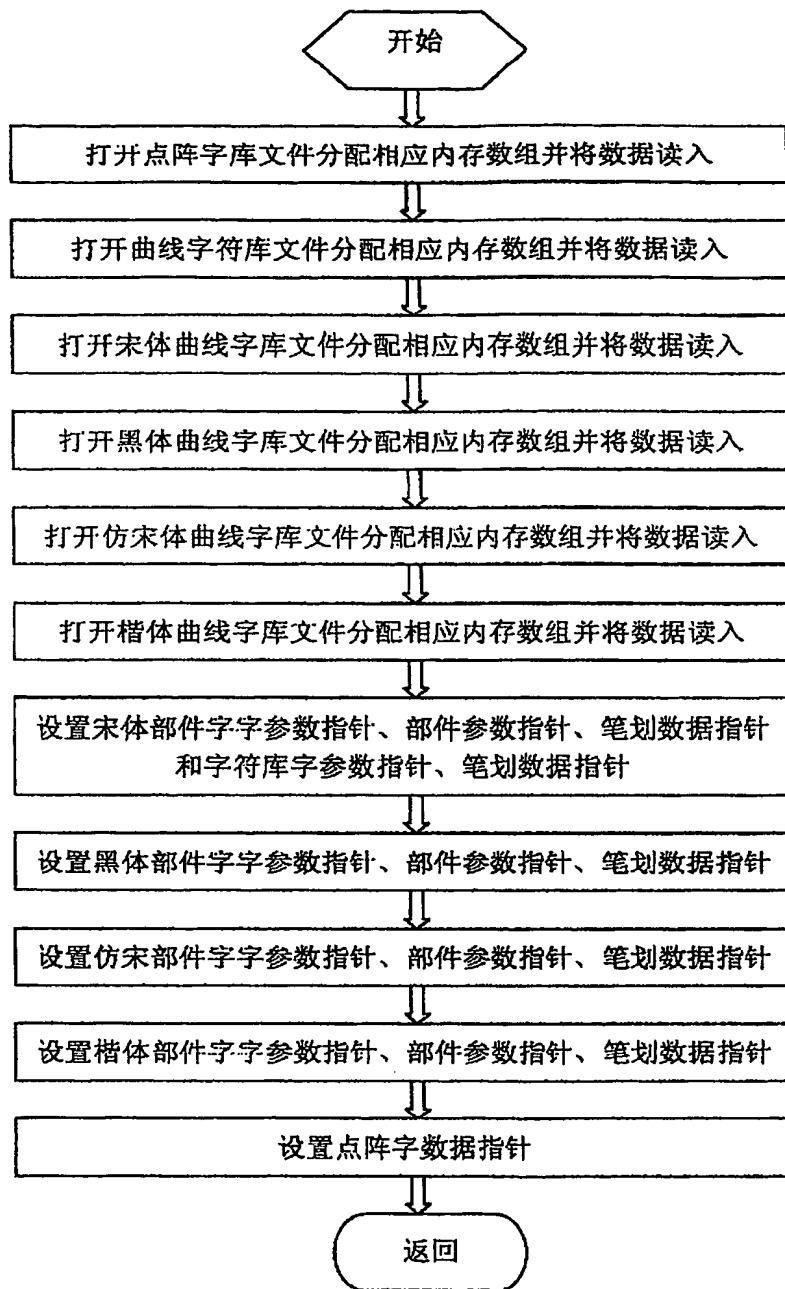


图 2

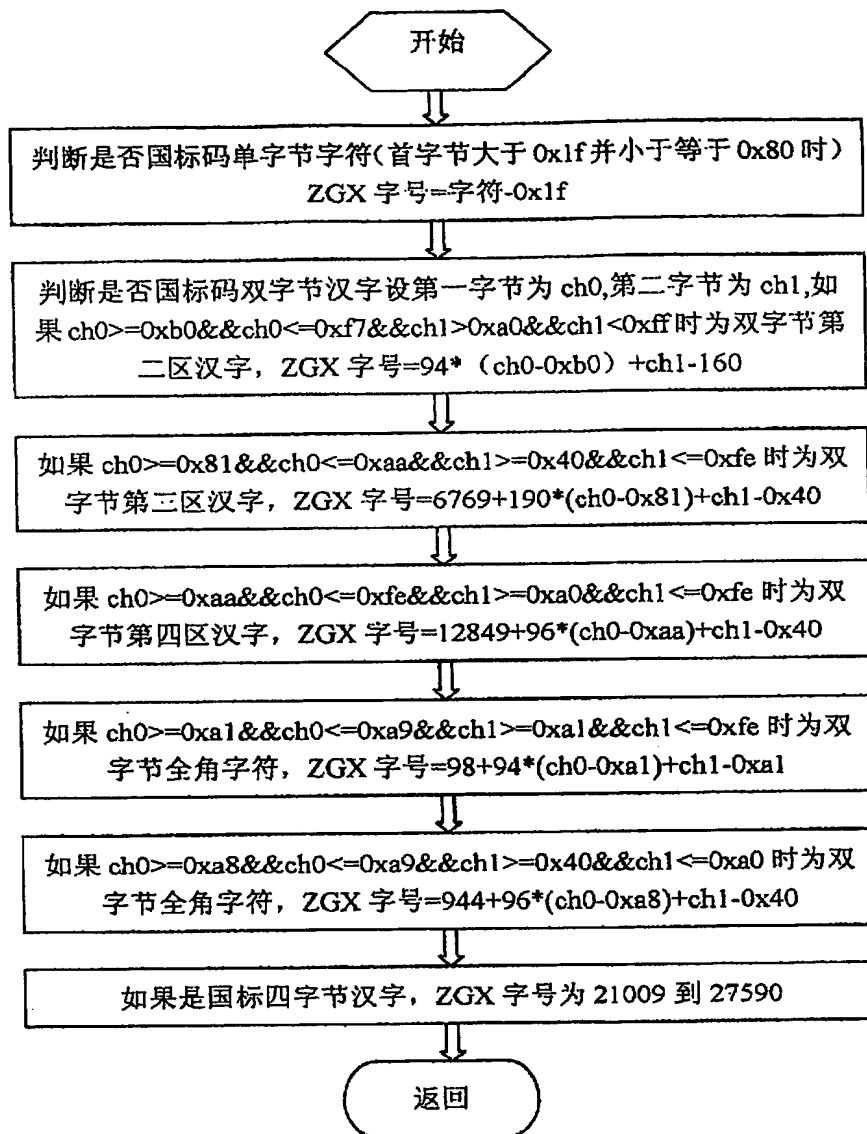


图 3

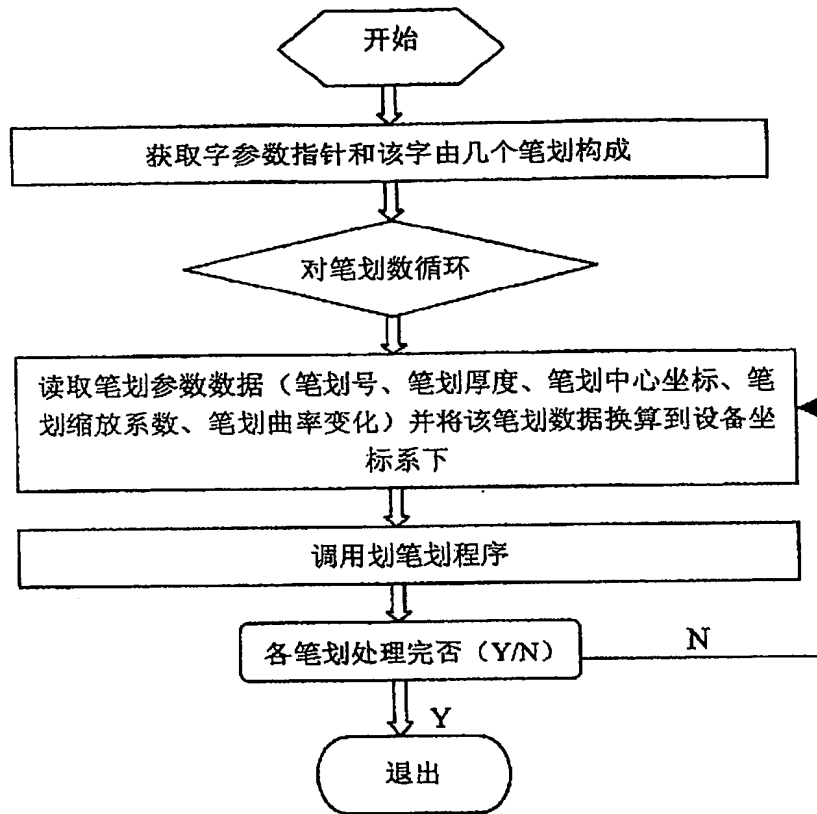
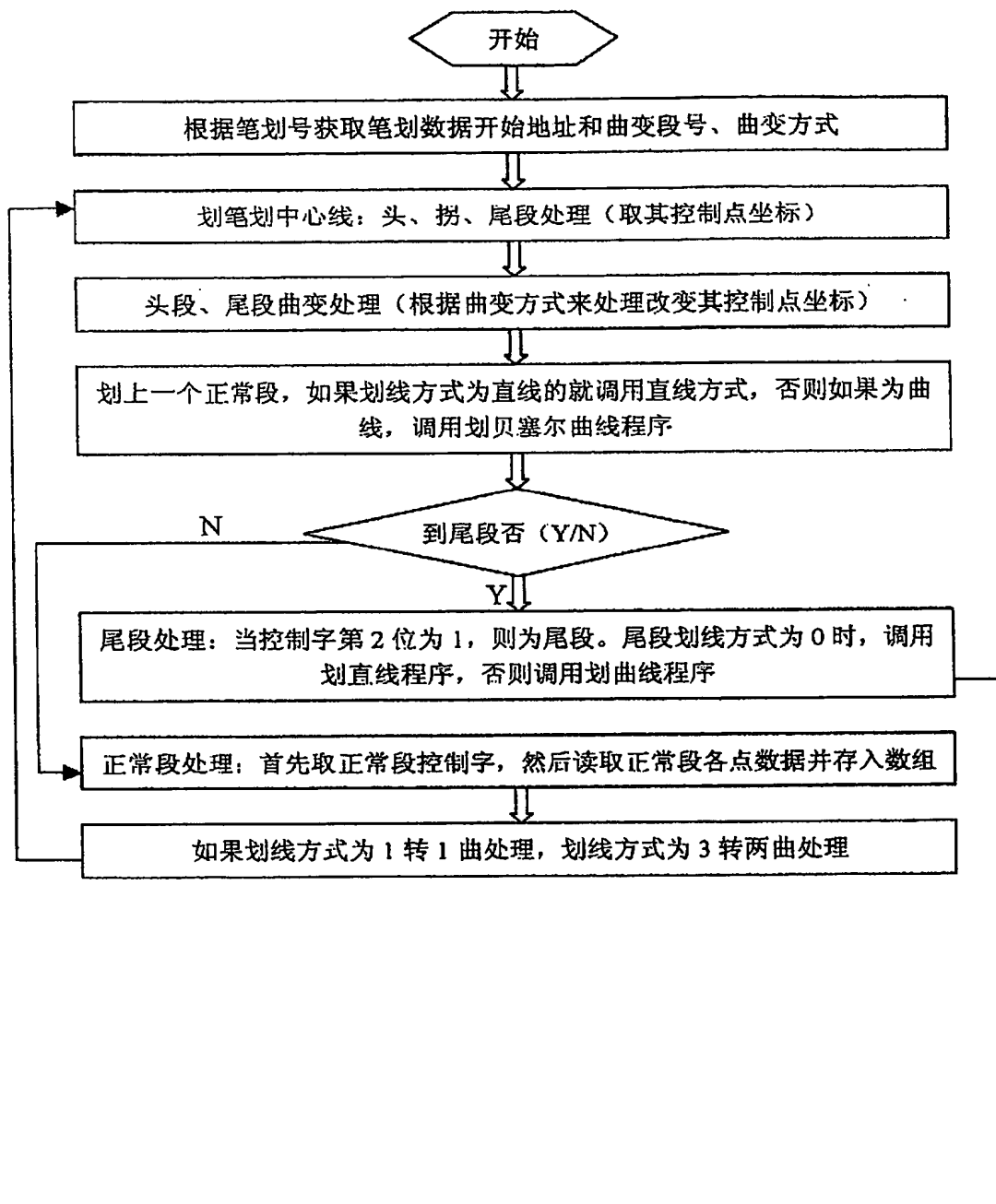


图 4



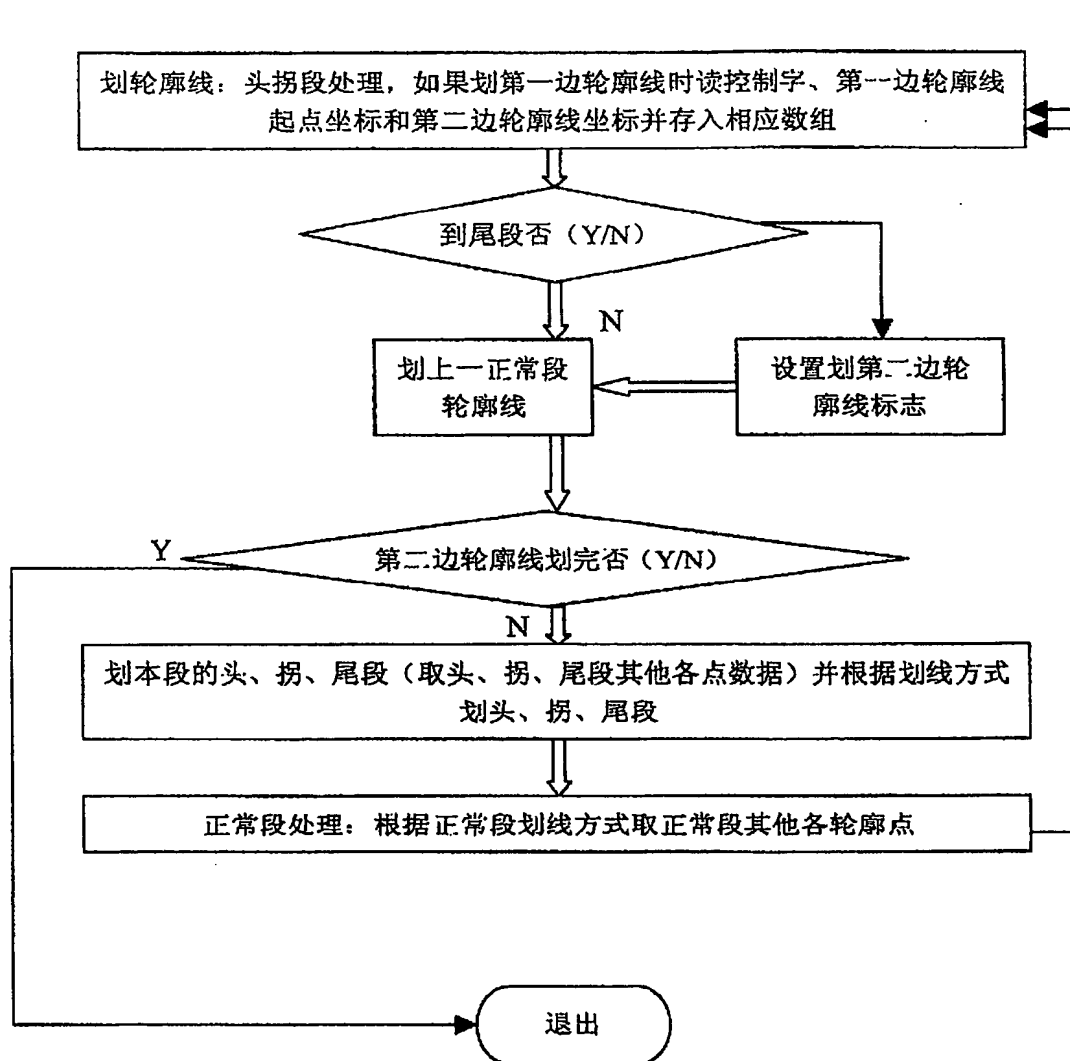


图 5

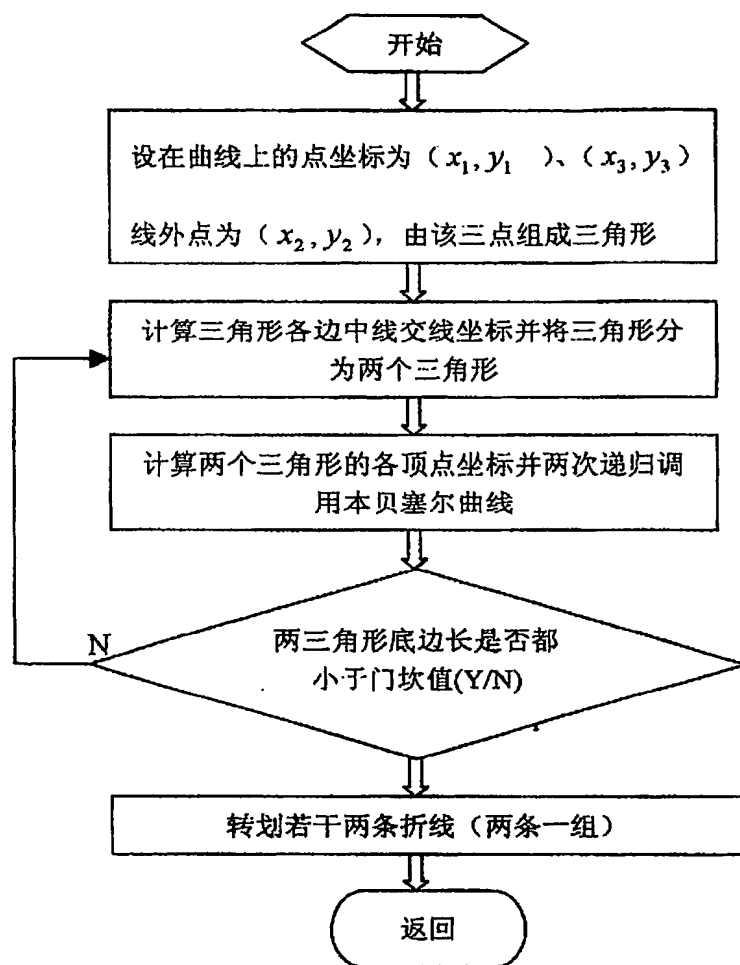


图 6

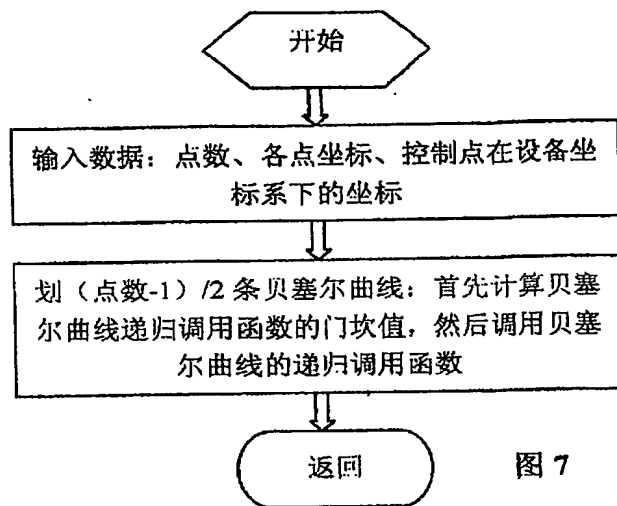


图 7

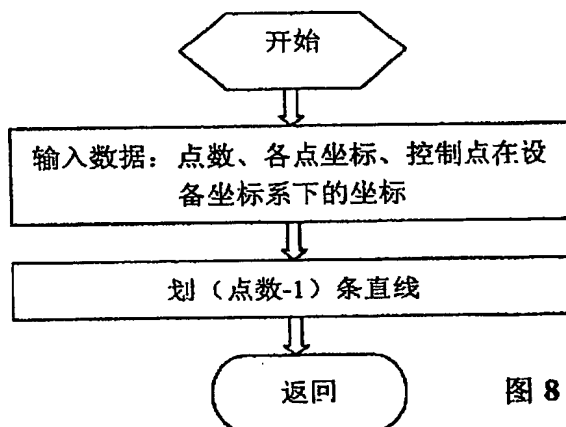


图 8

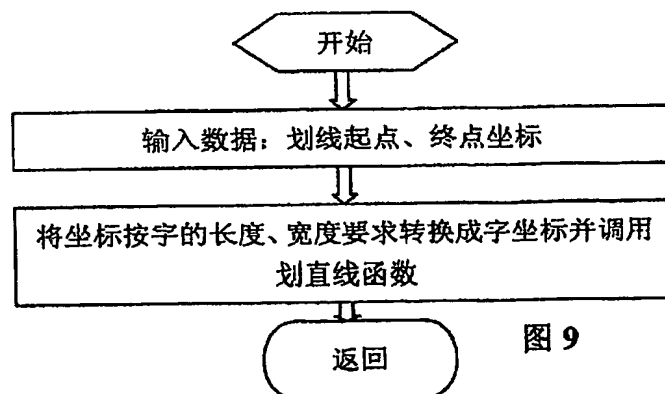


图 9



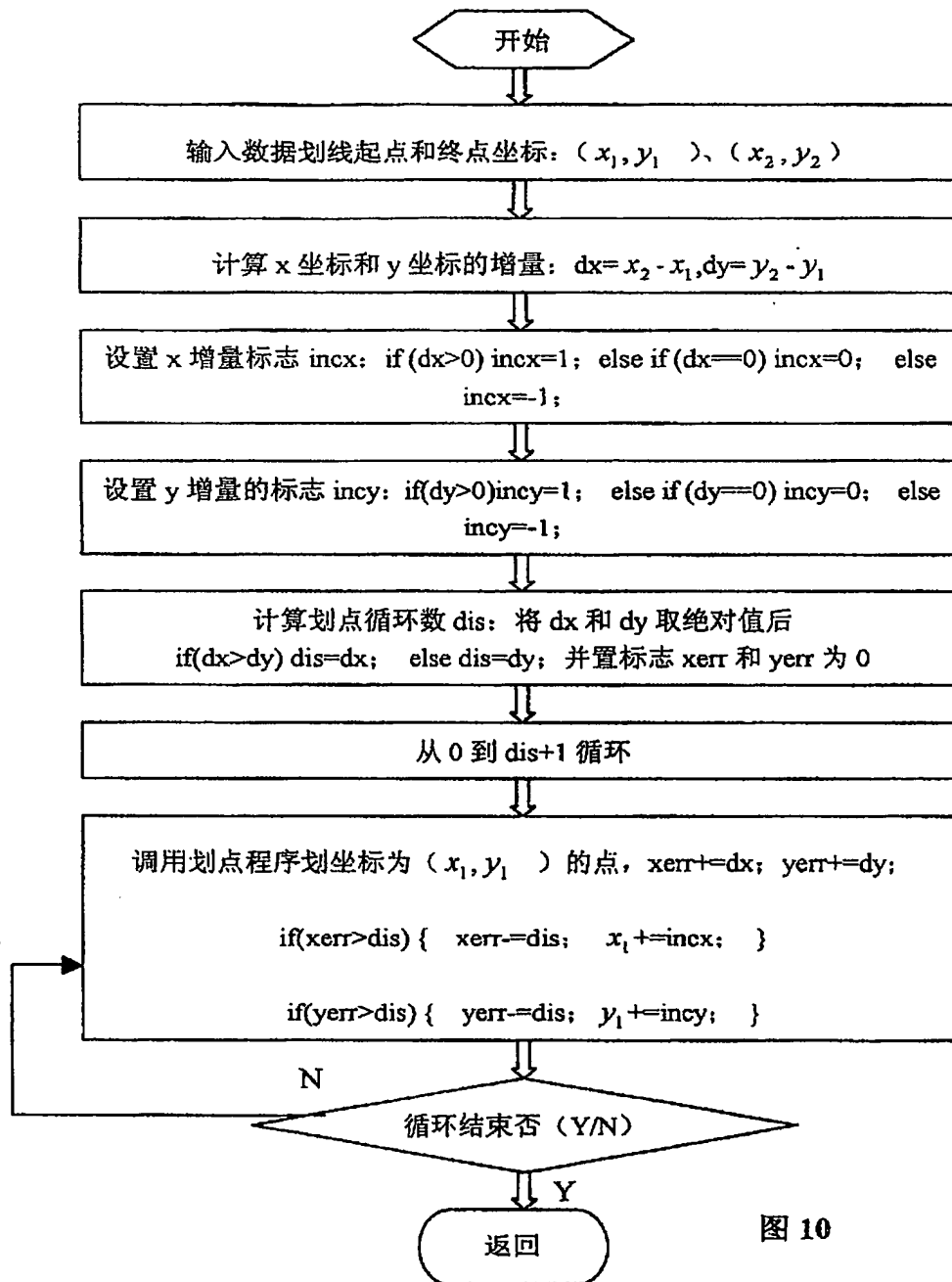
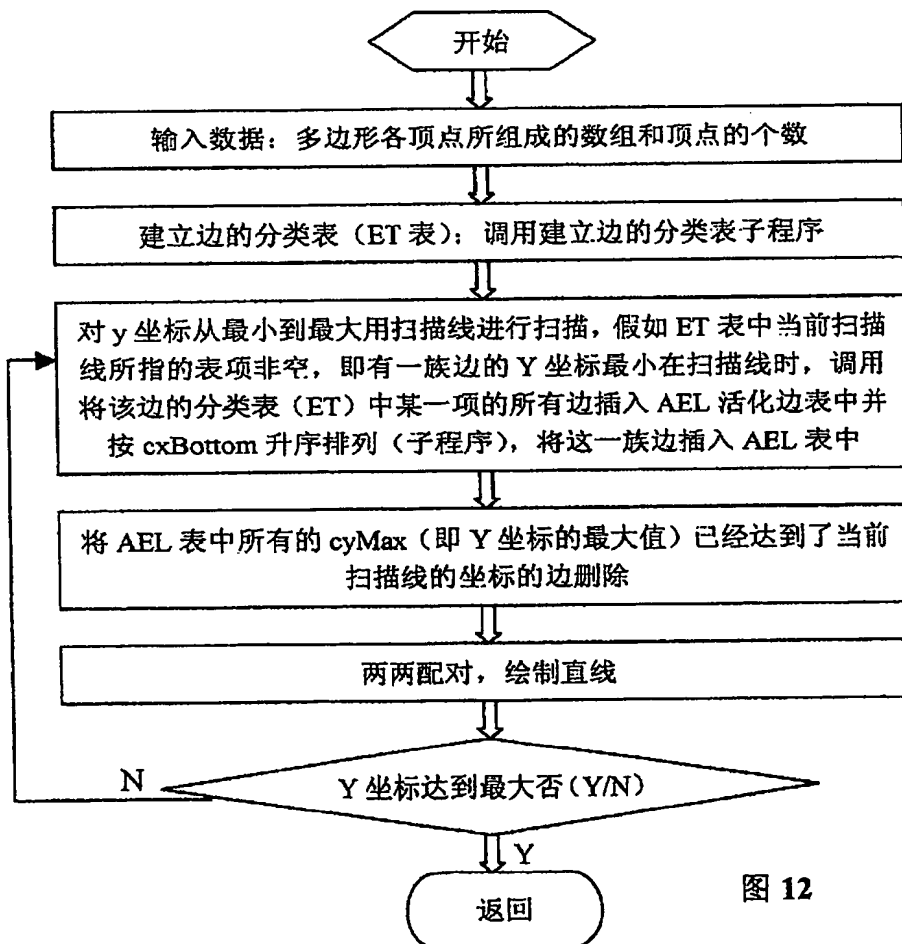
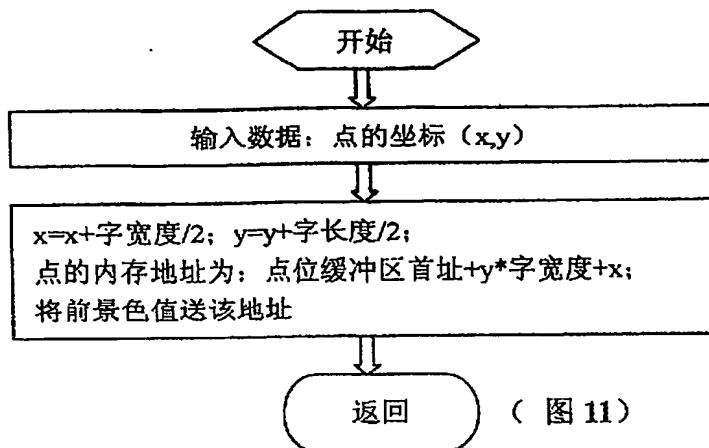


图 10



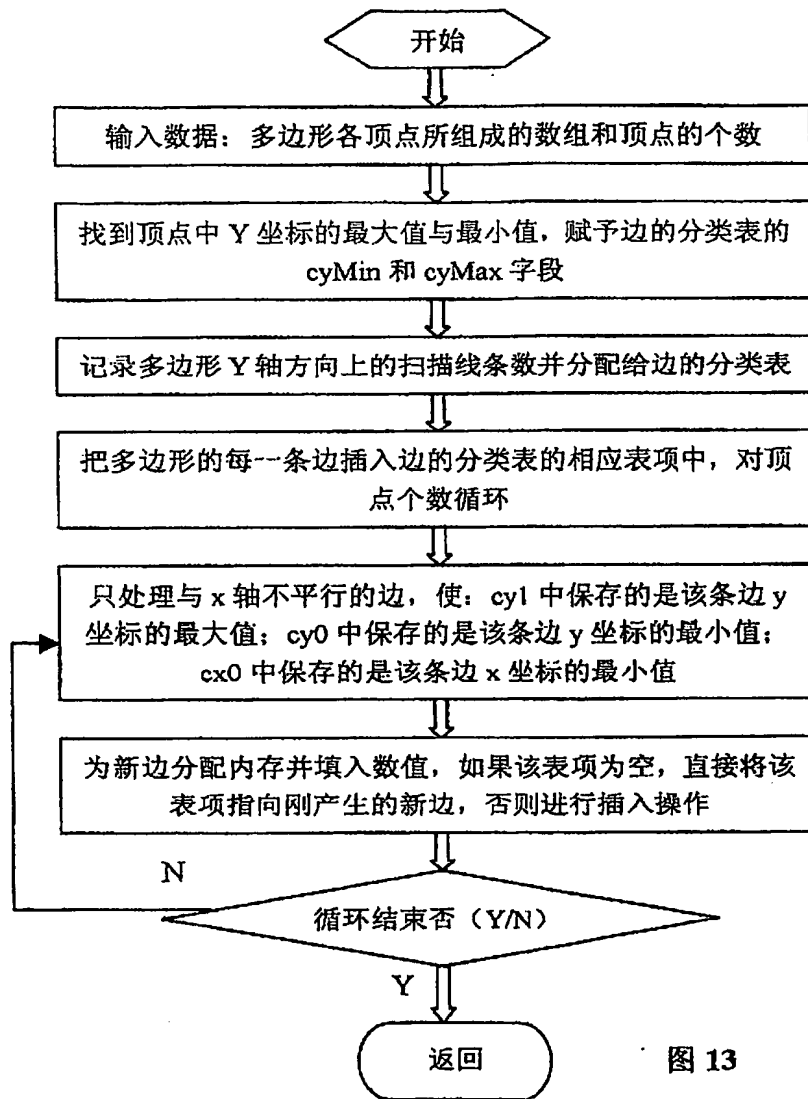


图 13

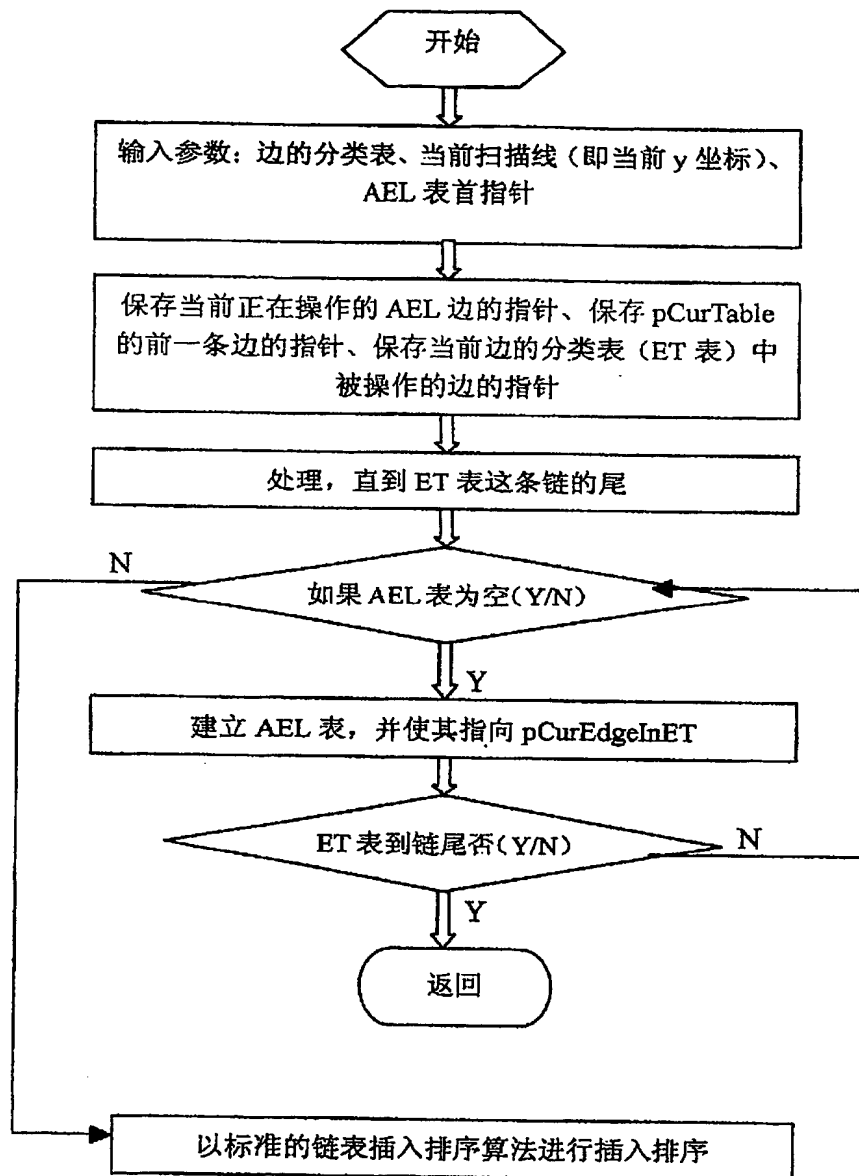


图 14